12-26-06

ZRWAF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE	
APPEAL BRIEF	Atty. Docket No. **VIGN1200-1**

Applicant **Conleth S. O'Connell**	
Application Number **09/965,914**	Date Filed **Sept. 28, 2001**
Title **Method and System for Cache Management for Dynamically-Generated Content (as amended)**	
Group Art Unit **2141**	Examiner **Luu, Le H.**
Confirmation Number: **4230**	

Mail Stop: Appeal Brief- Patents

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

Dear Sir:

Certification Under 37 C.F.R. §1.10

I hereby certify that this document is being deposited with the United States Postal Service with sufficient postage as Express Mail to Addressee (Label No. **EV964340505US**) in an envelope addressed to: Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313 on December 22, 2006.

Janice Pampell

Janice Pampell

Further to the Notice of Appeal filed August 11, 2006, the Appellant presents this Appeal Brief. The Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

12/27/2006 HVUONG1 00000007 09965914
01 FC:1402 500.00 OP

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The subject application is owned by Vignette, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 1301 S. MoPac Expressway, Suite 100, Austin, TX 78746.

II. RELATED APPEALS AND INTERFERENCES

The Appellant believes that there are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-61 were originally filed with the application. The claims were rejected in office actions dated March 25, 2005, November 23, 2005 and April 11, 2006. The claims have not been amended and stand rejected under 35 U.S.C. §103(a) as being obvious in light of the combination of United States Patent No. 6, 591,266 issued to Li ("Li") and United States patent No. 6,696,849 issued to Carlson ("Carlson"). Claims 1-61 are being appealed herein.

IV. STATUS OF AMENDMENTS

Claims 1-61 were filed with the original application and have not been amended during prosecution. Claims 1-61 are under Appeal.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Independent Claims 1, 23 and 43 are involved in the present appeal. The Appellant respectfully submits the following concise explanation of the subject matter defined in each of the independent Claims 1, 23 and 43.

Claim 1 recites:

> A method for cache management and regeneration of dynamically-generated content ("DGC") in one or more server computers within a client-server computer network, comprising the steps of:
>
> in response to a regeneration event, identifying a set of one or more previously cached DGC components affected by said regeneration event;
>
> regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event; and
>
> replacing each affected DGC component in said set with said respective new version of each.

Claim 1 recites a method for managing a cache and regeneration of dynamically-generated content ("DGC") in one or more server computers within a client-server computer network. As shown in FIGRUE 1, a server computer 24 in client-server computer network 20. See Application, page 13, lines 9-13. Server computer 24 can include a page generator program 50 that can dynamically generate files in response to file requests from client computer 22. See id. at page 15, lines 12-23. The content generated by page generator program can be cached by plug-in 50. See id. at page 15, lines 23-26.

The method includes the step of "in response to a regeneration event, identifying a set of one or more previously cached DGC components affected by said regeneration event." When an event occurs that indicates that previously generated dynamic content may be out of date or may otherwise need to be updated, the previously generated dynamically-generated content affected by the event can be identified. For example, if a template from which dynamic content was previously generated changes, the previously generated content from that template is identified (e.g., using the docroot file system 72 or other mechanism). *See* page 21, lines 14-27.

The method further includes the steps of "regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event." Regenerated content incorporate criteria associated with the regeneration event such that the regenerated page is different than the previous version of the page. For example, a change to a template can cause regeneration of pages generated by that template to incorporate the change. See at page 21, lines 21-25.

The method further includes "replacing each affected DGC component in said set with said respective new version of each." Thus, the superseded versions of the customized pages affected by the regeneration event are atomically replaced with the newly generated pages. *See* id. at page 22, lines 5-9.

Claim 23 recites:

> A system for cache management and regeneration of dynamically-generated content ("DGC") in one or more server computers within a client-server computer network, comprising:
> instructions for, in response to a regeneration event, identifying a set of one or more previously cached DGC components affected by said regeneration event;
> instructions for regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event; and
> instructions for replacing each affected DGC component in said set with said respective new version of each.

Claim 23 recites a system for cache management and regeneration of dynamically-generated content ("DGC") in one or more server computers within a client-server computer network. As shown in FIGRUE 1, a server computer 24 in client-server computer network 20. See Application, page 13, lines 9-13. Server computer 24 can include a page generator program 50 that can dynamically generate files in response to file requests from client computer 22. See id. at page 15, lines 12-23. The content generated by page generator program can be cached by plug-in 50. See id. at page 15, lines 23-26.

Claim 23 further recites "instructions for, in response to a regeneration event, identifying a set of one or more previously cached DGC components affected by said regeneration event." When an event occurs that indicates that previously generated dynamic content may be out of date or may otherwise need to be updated, the previously generated dynamically-generated content affected by the event can be identified. For example, if a template from which dynamic

content was previously generated changes, the previously generated content from that template is identified (e.g., using the docroot file system 72 or other mechanism). See page 21, lines 14-27.

Claim 23 further recites "instructions for regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event." Regenerated content incorporate criteria associated with the regeneration event such that the regenerated page is different than the previous version of the page. For example, a change to a template can cause regeneration of pages generated by that template to incorporate the change. See at page 21, lines 21-25.

Claim 23 further recites "replacing each affected DGC component in said set with said respective new version of each." As described in the Application, superseded versions of the customized pages affected by the regeneration event are atomically replaced with the newly generated pages. See id. at page 22, lines 5-9.

Claim 43 recites:

> A method for cache management and regeneration of dynamically-generated content ("DGC) in one or more server computers within a client-server computer network, comprising the steps of:
> initiating a regeneration event to affect one or more previously cached DGC components;
> in response to said regeneration event, identifying a set of one or more of said previously cached DGC components affected by said regeneration event;
> regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event; and
> replacing each affected DGC component in said set with said respective new version of each.

As shown in FIGRUE 1, a server computer 24 in client-server computer network 20. See Application, page 13, lines 9-13. Server computer 24 can include a page generator program 50 that can dynamically generate files in response to file requests from client computer 22. See id. at page 15, lines 12-23. The content generated by page generator program can be cached by plug-in 50. See id. at page 15, lines 23-26.

The method includes the step of "initiating a regeneration event to affect one or more

previously cached DGC components." A regeneration event can be initiated by modifying a template, initiating a flushing operation or taking other action. *See* id. at page 21, lines 17-19.

The method further includes the step of "in response to said regeneration event, identifying a set of one or more previously cached DGC components affected by said regeneration event." When an event occurs that indicates that previously generated dynamic content may be out of date or may otherwise need to be updated, the previously generated dynamically-generated content affected by the event can be identified. For example, if a template from which dynamic content was previously generated changes, the previously generated content from that template is identified (e.g., using the docroot file system 72 or other mechanism). *See* page 21, lines 14-27.

The method further includes the steps of "regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event." Regenerated content incorporate criteria associated with the regeneration event such that the regenerated page is different than the previous version of the page. For example, a change to a template can cause regeneration of pages generated by that template to incorporate the change. See at page 21, lines 21-25.

The method further includes "replacing each affected DGC component in said set with said respective new version of each." Thus, the superseded versions of the customized pages affected by the regeneration event are atomically replaced with the newly generated pages. *See* id. at page 22, lines 5-9.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

In the April 11, 2006 Office Action (the "April 11 Office Action"), the Examiner rejected Claims 1-61 under 35 U.S.C. §103 (a) as being unpatentable over United States Patent No. 6,591,266 ("Li") in view of United States Patent No. 6,697,879 ("Carlson"). In rejecting Claims 1-61, the Examiner applied an inappropriate standard and failed to consider the merits of a 37 C.F.R. 1.131 Declaration as showing conception. The grounds for rejection to be reviewed are whether the Examiner erred in determining that the facts of the 37 C.F.R. 1.131 Declaration do not show conception such that the date of conception of the present invention is prior to the effective date of the Li reference.

VII. ARGUMENT

REJECTIONS UNDER 35 U.S.C. §103(a)

1. Claims 1-61

In an Office Action dated April 11, 2006 (the "April 11 Office Action"), the Examiner rejected Claims 1-61 under 35 U.S.C. §103(a) as being obvious in light of the combination of Li and Carlson. The Examiner had previously rejected the same claims over the same art in an Office Action dated March 25, 2005 (the "March 25 Office Action") and an Office Action dated November 23, 2005 (the "November 23 Office Action").

Li issued on July 8, 2003, after the present application was filed and therefore is presumably being considered prior art under 35 U.S.C. 102(e) for purposes of the rejection under 35 U.S.C. §103(a). According to 35 U.S.C. 102(e) the invention must be described in "an application for patent or . . . a patent granted on an application for patent by another filed in the United States before the invention by the Appellant for patent." Li claims priority to United States Provisional Patent Application No. 60/218,418 having a filing date of July 14, 2000. Therefore, the earliest effective filing date of Li is July 14, 2000 (assuming that '418 Provisional supports Li, an issue not currently being argued). To properly qualify as prior art under 35 U.S.C. 102(e), the July 14, 2000 effective filing date of Li must be earlier than the date of invention of the present invention. Part of showing invention prior to the effective filing date of Li is showing conception prior to the effective filing date of Li. In responding to the various Office Actions, the Appellants submitted a Declaration Under 37 C.F.R. 1.131 (the "Declaration") to this end. The Declaration avers to the fact that the present invention was conceived at least as early as January 7, 2000 and the corresponding Office Actions state that constructive reduction to practice occurred on September 29, 2000 with the filing of the provisional application. The Examiner has failed to consider the merits of the Declaration because the Examiner applied an inappropriate standard, apparently looking for proof of reduction to practice rather than conception.

The prosecution record of the present application reveals that the Examiner, without cause, did not believe the veracity of the Declaration and applied an inappropriate standard in

reviewing the Declaration. On August 25, 2005, Appellant submitted a reply to the March 25, 2005 Office Action (the "August 25 Reply"). Attached to the August 25 Reply was the Declaration, including an email drafted by Mr. O'Connell as Exhibit A and a software specification distributed by the inventors that described in the claimed invention as Exhibit B. A copy of the Declaration, Exhibit A and Exhibit B are included in the Evidence Appendix. Mr. O'Connell averred to the following facts: the claimed invention was conceived at least as early as January 7, 2000 (*see* Declaration, page 1, ¶2); the email attached as Exhibit A to the Declaration was drafted on January 7, 2000 (*see* Declaration, page 1, ¶3); the specification attached to Exhibit B of the declaration was drawn up prior to January 7, 2000 (*see* Declaration, page 1 ¶4); and the specification attached as Exhibit B demonstrates conception of the invention as described and claimed in the patent application, (*see* Declaration, page 1 ¶5). Mr. O'Connell acknowledged that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and may jeopardize the validity of the application or any patent issuing thereon. (*see* Declaration, page 1 ¶6). The August 25 Reply also states that the invention was reduced to practice by filing of a provisional application on September 29, 2000. Thus, the Declaration is used to establish a conception date while the filing date of the provisional application establishes the date of a constructive reduction to practice.

In the November 23 Office Action, the Examiner stated that "the evidence submitted is insufficient to establish conception . . . Exhibits A and B do not explicitly prove, demonstrate, nor clearly show in details how the claimed invention can constructed using information from the email and outline in Exhibits A and B." *See* November 23 Office Action, page 2. The Examiner then went on to question the truth of Mr. O'Connell's statements that the email was sent on January 7, 2000, stating "the validity of the email in Exhibit A is questionable because Examiner notes that [sic] time stamp and subject header of the email in Exhibit A were missing."

Appellant again submitted the Declaration in a Reply dated February 23, 2005 (the "February 23 Reply"). In the April 11, 2006 Office Action, the Examiner again rejected the Declaration stating:

> The evidence submitted is insufficient to establish a conception of the invention prior to the effective date of the Li et al. reference . . . conception is more than a vague idea of how to solve a problem. The requisite

> means themselves and their interaction must also be comprehended . . . Exhibits A and B do not explicitly prove, demonstrate, nor clearly show in details how the claimed invention can be constructed using information from the email and outline in the Exhibits A and B . . . The affidavit or declaration and exhibits must clearly explain which facts or data Appellant is relying on to show completion of his or her invention prior to the particular date. Vague and general statements in broad terms about what exhibits describe along with a general assertion that the exhibits describe a reduction to practice 'amounts to mere pleading, unsupported by proof or showing of facts' and, thus does not satisfy the requirements of 37 C.F.R. 1.131 (b). *See*, April 11 Office Action, page 2 [Emphasis Added].

The Examiner's rejection of the Declaration was improper and the Examiner's baseless questioning of the veracity of Mr. O'Connell's statements regarding the date of the email—statements made in a legal document in which Mr. O'Connell acknowledged that willful false statements were punishable by fine or imprisonment—was inappropriate.

First addressing the Examiner's inappropriate questioning of Mr. O'Connell's averments, Appellant points to MPEP 715.07, which clearly states "if the dates of the exhibits have been removed or blocked off, the matter of dates can be taken care of in the body of the oath or declaration." Thus, Mr. O'Connell's averments in the Declaration establish that the email was sent on January 7, 2000 are sufficient to establish that the email was sent January 7, 2000 and there is no basis to question the veracity of this statement and it is improper to do so.

Turning now to the propriety of the Examiner's rejection, it appears that the Examiner is seeking unnecessary corroboration for the inventor's statements with respect to conception. Moreover, the Examiner is constantly looking for evidence as to reduction to practice—a different issue than conception—to prove conception. Indeed, the Examiner even quotes passages of the MPEP that are drawn to evidence regarding reduction to practice, not conception, specifically suggesting the Declaration must show "facts or data applicant is relying on to show completion of his or her invention prior to the particular date." MPEP 715.07. Reduction to practice, however, is not required to show conception and is not at issue in the Declaration. The date of Reduction to practice relied upon for purpose of the August 25 and February 23 Replies is the date of filing of the provisional application as stated in the August 25

Reply and subsequent replies. *See e.g., August 25 Reply, page 5.* Thus, the Examiner fails to consider the merits of the Declaration with respect to conception based on what appears to be the belief that the Declaration is attempting to show reduction to practice.

Appellant notes, that unlike interference practice, "averments made in a 37 C.F.R. 1.131 affidavit or declaration do not require corroboration; an inventor may stand on his or her own affidavit if he or she so elects." *See* MPEP 715.07 (emphasis added). In this case, Mr. O'Connell has averred to the fact that he conceived the claimed invention at least as early as January 7, 2000 and that the conception of the invention is demonstrated in Exhibit B of his declaration. *See* August 25 Reply, Exhibit 1, page 1, ¶2, 4, 5. Moreover, Mr. O'Connell has provided more than necessary as he has provided corroborating evidence, in the form of Exhibit B of the Declaration, demonstrating conception of this invention.

Moreover, "in reviewing a 37 C.F.R. 1.131 affidavit or declaration, the examiner must consider all the evidence presented in its entirety." See MPEP 715.07 (emphasis added). In addition to Mr. O'Connell's averments, the evidence as a whole shows that the inventors had "more than a vague idea of how to solve a problem." The email of Exhibit A of the Declaration evidences that the inventors distributed the specification of Exhibit B for purposes of implementation, demonstrating that the inventors had much more than "a mere vague idea of how to solve the problem . . ." Specifically, the email, in discussing the specification of Exhibit B of the Declaration, states, "I'm hoping it is clear enough. If not I'm available to talk about. If any other issues come up as you go through the implementation . . . ." *See* Declaration, Exhibit A (referring to Exhibit B in which Mr. O'Connell provides a detailed specification showing conception of the invention).

The evidence as a whole, including the averments of Mr. O'Connell in the Declaration and the Exhibits attached to the Declaration show that, on January 7, 2000, the inventors had more than a "vague idea of how to solve the problem" but had at least reached the point of distributing the specification for purposes of implementation. Thus, the Declaration and attached exhibits clearly demonstrate conception at least as early as January 7, 2000. Therefore, the Examiner should examine the application during the course of prosecution based on the fact that conception occurred at least as early as January 7, 2000.

2. Conclusion

The Declaration and attached exhibits clearly demonstrate conception at least as early as January 7, 2000. The Examiner's repeated rejections based on the belief that the Declaration fails to show conception should be withdrawn.

A check in the amount of $500.00 is included with this filing. While Appellant believes no further fees are due and owing, if Appellant is in error, the Commissioner is hereby authorized to deduct the appropriate amount from Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully submitted,
Sprinkle IP Law Group

John L. Adair
48,828

Date: December 22, 2006

1301 W. 25th Street, Suite 408
Austin, TX 78705
Tel. (512) 637-9220
Fax. (512) 371-9088

VIII. CLAIMS APPENDIX

1. (Original) A method for cache management and regeneration of dynamically-generated content ("DGC") in one or more server computers within a client-server computer network, comprising the steps of:

in response to a regeneration event, identifying a set of one or more previously cached DGC components affected by said regeneration event;

regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event; and

replacing each affected DGC component in said set with said respective new version of each.

2. (Original) The method of Claim 1, further comprising the step of serving said new version of one or more of said affected DGC components to a client computer in said client-server network in response to a request from said client computer.

3. (Original) The method of Claim 1, further comprising the step of serving said new version of one or more of said affected DGC components in the form of a dynamically-generated page to a client computer in said client-server network in response to a request from said client computer.

4. (Original) The method of Claim 1, wherein:

said identifying step further comprises identifying which of said affected DGC components satisfy a threshold criteria;

said set of affected DGC components comprises only those affected DGC components that satisfy said threshold criteria; and

said replacing step further comprises flushing those of said affected previously cached DGC components that do not satisfy said threshold criteria.

5. (Original) The method of Claim 4, wherein said threshold criteria is an arbitrary value of an arbitrary parameter.

6. (Original) The method of Claim 5, wherein said arbitrary parameter is an elapsed time since the last client computer request for a DGC component or for a dynamically-generated page.

7. (Original) The method of Claim 1, wherein any one or more of said identifying, regenerating and replacing steps can be performed at a different one of said one or more server computers from each other.

8. (Original) The method of Claim 1, wherein said regenerating step further comprises the step of limiting to a preset threshold value the number of affected DGC component regenerations that can simultaneously occur.

9. (Original) The method of Claim 8, wherein said preset threshold value is arbitrarily determined according a desired network performance level.

10. (Original) The method of Claim 8, wherein said preset threshold value is determined by a static descriptor, such as a configuration variable.

11. (Original) The method of Claim 1, wherein said regeneration event comprises a change to a page template, an explicit flushing event, or a change to a DGC component.

12. (Original) The method of Claim 11, wherein said explicit flushing event comprises the expiration of a preset time period.

13. (Original) The method of Claim 1, wherein said criteria associated with said regeneration event is a change to a page template from which one or more previously cached dynamically-generated pages ("DGPs") were generated.

14. (Original) The method of Claim 1, wherein said criteria associated with said regeneration event is a change to the content of one or more of said previously cached DGC components, or no criteria.

15. (Original) The method of Claim 1, wherein every cached DGC component is associated with a custom cached file name comprising a combination of an initial file request name with a selected attribute of a computer user.

16. (Original) The method of Claim 15, wherein said selected attribute is selected from the group including browser name, user language, computer domain, computer platform, and content ID.

17. (Original) The method of Claim 15, wherein said selected attribute is a default attribute.

18. (Original) The method of Claim 17, wherein said default attribute is no user attribute.

19. (Original) The method of Claim 15, wherein said selected attribute is used in said regenerating step to regenerate said new versions of said affected DGC components.

20. (Original) The method of Claim 15, wherein said selected attribute is keyed to a particular application.

21. (Original) The method of Claim 1, further comprising the step of updating a docroot file system to indicate changes resulting from replacing said affected DGC components.

22. (Original) The method of Claim 21, wherein said docroot file system is associated with a memory-based cache repository or a file-based cache repository.

23. (Original) A system for cache management and regeneration of dynamically-generated content ("DGC") in one or more server computers within a client-server computer network, comprising:

instructions for, in response to a regeneration event, identifying a set of one or more previously cached DGC components affected by said regeneration event;

instructions for regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event; and

instructions for replacing each affected DGC component in said set with said respective new version of each.

24. (Original) The system of Claim 23, further comprising instructions for serving said new version of one or more of said affected DGC components to a client computer in said client-server network in response to a request from said client computer.

25. (Original) The system of Claim 23, further comprising instructions for serving said new version of one or more of said affected DGC components in the form of a dynamically-generated page ("DGP") to a client computer in said client-server network in response to a request from said client computer.

26. (Original) The system of Claim 23, wherein:

said instructions for identifying further comprise instructions for identifying which of said affected DGC components satisfy a threshold criteria;

said set of affected DGC components comprises only those affected DGC components that satisfy said threshold criteria; and

said instructions for replacing further comprise instructions for flushing those of said affected previously cached DGC components that do not satisfy said threshold criteria.

27. (Original) The system of Claim 26, wherein said threshold criteria is an arbitrary value of an arbitrary parameter.

28. (Original) The system of Claim 27, wherein said arbitrary parameter is an elapsed time since the last client computer request for a DGC or for a DGP.

29. (Original) The system of Claim 23, wherein said instructions for regenerating further comprise instructions for limiting to a preset threshold value the number of affected DGC component regenerations that can simultaneously occur.

30. (Original) The system of Claim 29, wherein said preset threshold value is determined according a desired network performance level or according to a static descriptor, such as a configuration variable.

31. (Original) The system of Claim 23, wherein said regeneration event comprises a change to a page template, an explicit flushing event, or a change to a DGC component.

32. (Original) The system of Claim 23, wherein said criteria associated with said regeneration event is a change to a page template from which one or more previously cached DGPs were generated.

33. (Original) The system of Claim 23, wherein said criteria associated with said regeneration event is a change to the content of one or more of said previously cached DGC components.

34. (Original) The system of Claim 23, wherein said criteria associated with said regeneration event is no change.

35. (Original) The system of Claim 23, wherein every cached DGC component is associated with a custom cached file name comprising a combination of an initial file request name with a selected attribute of a computer user.

36. (Original) The system of Claim 35, wherein said selected attribute is selected from the group including browser name, user language, computer domain, computer platform, and content ID.

37. (Original) The system of Claim 35, wherein said selected attribute is used in said regenerating step to regenerate said new versions of said affected DGC components.

38. (Original) The system of Claim 35, wherein said selected attribute is not a user attribute.

39. (Original) The system of Claim 35, wherein said selected attribute is keyed to a particular application.

40. (Original) The system of Claim 23, further comprising instructions for updating a docroot file system to indicate changes resulting from replacing said affected DGC components.

41. (Original) The system of Claim 40, wherein said docroot file system is associated with a cache repository.

42. (Original) The system of Claim 41, wherein said cache repository is a file-based cache repository.

43. (Original) A method for cache management and regeneration of dynamically-generated content ("DGC) in one or more server computers within a client-server computer network, comprising the steps of:

initiating a regeneration event to affect one or more previously cached DGC components;

in response to said regeneration event, identifying a set of one or more of said previously cached DGC components affected by said regeneration event;

regenerating a new version of each affected DGC component in said set to incorporate a criteria associated with said regeneration event; and

replacing each affected DGC component in said set with said respective new version of each.

44. (Original) The method of Claim 43, wherein said regeneration event is initiated by a user via a user interface.

45. (Original) The method of Claim 44, wherein said user interface comprises a standard user-to-computer interface to access an interface program.

46. (Original) The method of Claim 43, wherein initiating said regeneration event comprises changing a template affecting one or more of said previously cached DGC components.

47. (Original) The method of Claim 43, wherein initiating said regeneration event comprises initiating a flushing operation.

48. (Original) The method of Claim 43, wherein initiating said regeneration event comprises initiating a flushing operation in response to a change in the content of one or more of said previously cached DGC components.

49. (Original) The method of Claim 43, further comprising the step of serving said new version of one or more of said affected DGC components to a client computer in said client-server network in response to a request from said client computer.

50. (Original) The method of Claim 43, further comprising the step of serving said new version of one or more of said affected DGC components in the form of a dynamically-generated page ("DGP") to a client computer in said client-server network in response to a request from said client computer.

51. (Original) The method of Claim 43, wherein:

said identifying step further comprises identifying which of said affected DGC components satisfy a threshold criteria;

said set of affected DGC components comprises only those affected DGC components that satisfy said threshold criteria; and

said replacing step further comprises flushing those of said affected previously cached DGC components that do not satisfy said threshold criteria.

52. (Original) The method of Claim 51, wherein said threshold criteria is an arbitrary value of an arbitrary parameter.

53. (Original) The method of Claim 52, wherein said arbitrary parameter is an elapsed time since the last client computer request for a DGC component or a DGP.

54. (Original) The method of Claim 43, wherein any one or more of said initiating, identifying, regenerating and replacing steps can be performed at a different one of said one or more server computers from each other.

55. (Original) The method of Claim 43, wherein said regenerating step further comprises the step of limiting to a preset threshold value the number of affected DGC component regenerations that can simultaneously occur.

56. (Original) The method of Claim 43, wherein said criteria associated with said regeneration event is a change to a page template from which one or more previously cached DGPs were generated.

57. (Original) The method of Claim 43, wherein said criteria associated with said regeneration event is a change to the content of one or more of said previously cached DGC components, or no criteria.

58. (Original) The method of Claim 43, wherein every cached DGC component is associated with a custom cached file name comprising a combination of an initial file request name with a selected attribute of a computer user.

59. (Original) The method of Claim 58, wherein said selected attribute is selected from the group including browser name, user language, computer domain, computer platform, and content ID.

60. (Original) The method of Claim 58, wherein said selected attribute is used in said regenerating step to regenerate said new versions of said affected DGC components.

61. (Original) The method of Claim 43, further comprising the step of updating a docroot file system to indicate changes resulting from replacing said affected DGC components.

IX. EVIDENCE APPENDIX

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE	
Declaration Under 37 C.F.R. 1.131	Atty. Docket No. **VIGN1200-1**

Applicants **Conleth S. O'Connell, et al.**	
Application Number **09/965,914**	Date Filed **09/28/2001**
Title **System and Method for Cache Management for Dynamically-Generated Content**	
Group Art Unit **2141**	Examiner **Luu, Le Hien**
Confirmation Number: **4230**	

Certificate of Mailing Under 37 C.F.R. §1.8

I hereby certify that this document is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22312-1450 on August 25, **2005.**

Onita Cannon

1. I, Conleth S. O'Connell, am an original inventor of the invention described and claimed in the above-referenced patent application.

2. The invention claimed in the above-referenced patent application was conceived at least as early as January 7, 2000.

3. I drafted and sent the email attached as Exhibit A on January 7, 2000.

4. To the email of January 7, 2000 I attached a specification drawn up by myself and Mark Scheevel on or prior to January 7, 2000. Mark Scheevel is another original inventor of the invention described and claimed in the above referenced patent application. The specification attached to the email of January 7, 2000 (Exhibit A) is attached hereto as Exhibit B.

5. The specification attached hereto as Exhibit B demonstrates a conception of the invention described and claimed in the above-referenced patent application at least as early as the date of the email: January 7, 2000.

6. Declarant acknowledges that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and may jeopardize the validity of the application or any patent issuing thereon.

I, Conleth S. O'Connell, aver that all statements made of my own knowledge are true and all statements made on information and belief are believed to be true.

Conleth S. O'Connell
August 12, 2005

EXHIBIT A

From: Conleth O'Connell <cso@vignette.com>
To: mjm@vignette.com, mark@vignette.com, marjm@vignette.com
CC: dls@vignette.com

Mike, Mark M.,

I'm attaching the "spec" Mark S. and I pulled together. Mark M/ has already read through it once, so I'm hoping it is clear enough. If not I'm available to talk about it. If any other issues come up as you go through the implementation (short-term vs. what is needed in the product), feel free to modify the document (use change bars), so we can review those things going forward.

What's important in this release is enough to support them without worrying about bcaps support (or language) on solaris. I'm trying to find out from robin if they are using any PZN functionality to help simplify the ctld protocol being duplicated.

If you have any questions, email me (here and home: cso@austin.rr.com), call me at home: 335-5080 or page me (873-6420)

Con

EXHIBIT B

Cache management regeneration design

Mark and Con, et al

1 Problem characterization

1.1 Ctld overload

1.2 Web server resource usage

1.3 Disk thrashing (web site weirdness b/c backup files used)

1.4 database hit b/c of the new requests

1.5 Underlying cause: no file in docroot to serve

2 Requirements

2.1 Must support clear_cache from templates

2.2 Must support flushcache program task requests

2.3 Must support template modifications

2.4 Must deal with template variations

2.5 Must know how to prune insignificant cached files

2.6 Must not put undue burden on ctld for regenerating cached files

2.7 Must handle same flush request that is an update to an existing request

2.8 Must work equally across all platforms and web servers

3 Proposal: Modify cache manager to regenerate cached files

3.1 In FlushFile

3.1.1 Save stat structure for passing to TakeAction()

3.1.2 If "IsValidCurl", capture char * result to pass to TakeAction()

3.1.3 Else, pass NULL for CURL string to TakeAction

3.2 Modify TakeAction() where it handles RENAME requests

3.2.1 If the last access time (using incoming stat structure) is outside the time window specified by either config param or protocol override, keep doing what we do now (plain old rename)

3.2.2 Otherwise, IF curl string passed in is not null:

3.2.2.1 reuse ParseCurl() (from clients/common/curlUtil.c) to get the pieces of the Curl file.

3.2.2.2 Need to further parse bcaps field to grab language field

3.2.2.3 Use bcaps field to map the hash value to the set of symbols out of the metafile (if it's "00", do a no-op)

3.2.2.3.1 On a template update/create, we have the physical symbols to be used in the page generator protocol b/c the metafile entry is part of what cmd is being told to update.

3.2.2.3.2 If not a template update (meaning, someone is using [CLEAR_CACHE] from within a template to flush the cache), we have to go to the metafile in the metafiles cache to lookup the symbols.

3.2.3 ELSE it's just a file, (the path was not a CURL, just a plain old file, e.g., index.html)

3.2.4 Call the function (generatePage()) to generate a request to the page generator and save a successful request to a tempfile

3.2.5 Perform rename from temp file to old filename

3.3 GeneratePage() needs to be added

3.3.1 Functionally equivalent to clients/common/generatePageImpl.cpp

3.3.2 Input: (*this is only for the ctld page generator*)

3.3.2.1 Need the filename (CURL, including template path, or plain file) modulo the docroot path, i.e., the URI

3.3.2.2 Needs the original Curl segment, if it is a CURL (pointer that went into TakeAction())

3.3.2.3 If it exists, needs the Content id out of the CURL

3.3.3 Output:

3.3.3.1 Two filenames: from and to

3.3.3.2 If there was an error along the way return the filename and its backup equivalent

3.3.3.3 Else return the temp filename generated and the original filename

3.3.4 Create an HTTP request to the page generator (*using clients/mt_nsapi/emulate404.c as a model, refer to http://hydra.vignette.com:8080/Interfaces/DOCUMENTS/Architecture_Info/ctld-protocol.html for explanation of the settings*)

3.3.4.1 Specify a POST to '/'

3.3.4.2 Add the HTTP header stuff

3.3.4.2.1 Our content-type is application/x-www-form-urlencoded

3.3.4.2.2 Don't forget the content-length that is the size of the entity body (following the double newline that separates headers from post bodies)

3.3.4.3 Append the following in URI encoding format (name=value&name2=value2...)

3.3.4.3.1 socket=true

3.3.4.3.2 cache=auto

3.3.4.3.3 if there was a curl, curl=/0,...

3.3.4.3.4 for all bcaps symbols, the format is browser=<symbol>&browser=<symbol>&...

3.3.4.3.5 path=<templatepath>

3.3.4.3.6 content-type=text/html (unless we have CMD_MIMETYPE setting, in which case we have to request for a mapping) really what's needed is test/html or application/octet-stream the page gen only cares about it for the purposes of inserting the magic string..., so here we can fudge. By this we mean if the generated page ends in htm or html (notice the lack of . so shtml can be caught as well), then specify text/html, else use application/octet-stream in case it's a graphic or some other file that doesn't get the magic string prepended.

3.3.4.3.7 For personalization settings, we need to pass the appropriate name-val pairs used by the CURL and COMPONENT commands. (autourl-tracking, etc.)

3.3.4.3.8 Add pzn_url_exclude from config info

3.3.4.3.9 template=<template id> (*this is optional*)

3.3.4.3.10 oid=<content id>

3.3.5 Number of simultaneous page generator requests should be throttled (to avoid overburdening)

3.3.5.1 Counting semaphore on page generator requests

3.3.6 Call the page generator

3.3.7 If the response is a 200 only (i.e., if it is an OK response), save it to a temp file, and return the temp filename

3.3.8 Else, return the orig filename and backup version of it

3.3.9 The threading model in the cache manager is such that per path a thread is performing the flush, so nothing too fancy needs to happen on the temp filename in the same directory as where you're performing the rename (that also limits the clashing possibilities)

4 Other thoughts

4.1 Use ACE when appropriate

4.2 Config settings added to this:

4.2.1 Number of simultaneous regeneration requests to perform

4.2.2 Time window for what determines old versus new; think of the setting is inclusive for new, and make it seconds so comparisons are easy against st_atime

4.3 What's needed in the short term (i.e. for the customer in question):

4.3.1 Don't worry about browser caps (template variations); the customer in question isn't using them

4.3.2 Don't worry about how a time window can be specified in the flush protocol (in fact, no changes to the incoming flush request protocol should be done)

4.3.3 Keep the HTTP headers simple for now.

4.3.4 Do what makes sense as in all cases, but think in terms of this code lasting for awhile

5 Issues

5.1 In 5.5, Cobra issue: Need to switch on template type to know which page generator (and which protocol) to use to make the request.

5.2 Need to optimize, file checking to leverage metafile data.

5.3 RENAME and DELETE aren't granular enough for our actions:

5.3.1 We need to distinguish between template destruction ("DESTROY") requiring cache cleanup versus don't bother with a backup file for "old" cached files ("DELETE")

5.4 Need to manage virtual hosting requests: at some point we need to know the domain prefix mapping, so we need the prefix piece...

5.5 Need to get server-side values for HTTP headers sent to page generator representing the Server-side CGI variables

5.6 CTLD configs must allow CMD to talk to it, so if you have allowable IPs and NAMEs restricted, this needs to be opened up to the CMD (or set of CMDs)

5.7 In the 5.x and above releases, new references will need to be added for CMD to continue to do this...

5.8 Do we care if it's for a component? Not yet...

EXHIBIT 2

-------- Original Message --------
Subject: "specification" for cache regeneration
Date: Fri, 07 Jan 2000 20:01:46 -0600
From: Conleth O'Connell <cso@vignette.com>
To: mkm@vignette.com, mark@vignette.com, markm@vignette.com
CC: dls@vignette.com

Mike, Mark M.,

I'm attaching the "spec" Mark S. and I pulled together. Mark M/ has already read through it once, so I'm hoping it is clear enough. If not I'm available to talk about it. If any other issues come up as you go through the implementation (short-term vs. what is needed in the product), feel free to modify the document (use change bars), so we can review those things going forward.

What's important in this release is enough to support them without worrying about bcaps support (or language) on solaris. I'm trying to find out from robin if they are using any PZN functionality to help simplify the ctld protocol being duplicated.

If you have any questions, email me (here and home: cso@austin.rr.com), call me at home: 335-5080 or page me (873-6420)

Con

Cache management regeneration design

Mark and Con, et al

1 Problem characterization

1.1 Ctld overload

1.2 Web server resource usage

1.3 Disk thrashing (web site weirdness b/c backup files used)

1.4 database hit b/c of the new requests

1.5 Underlying cause: no file in docroot to serve

2 Requirements

2.1 Must support clear_cache from templates

2.2 Must support flushcache program task requests

2.3 Must support template modifications

2.4 Must deal with template variations

2.5 Must know how to prune insignificant cached files

2.6 Must not put undue burden on ctld for regenerating cached files

2.7 Must handle same flush request that is an update to an existing request

2.8 Must work equally across all platforms and web servers

3 Proposal: Modify cache manager to regenerate cached files

3.1 In FlushFile

3.1.1 Save stat structure for passing to TakeAction()

3.1.2 If "IsValidCurl", capture char * result to pass to TakeAction()

3.1.3 Else, pass NULL for CURL string to TakeAction

3.2 Modify TakeAction() where it handles RENAME requests

3.2.1 If the last access time (using incoming stat structure) is outside the time window specified by either config param or protocol override, keep doing what we do now (plain old rename)

3.2.2 Otherwise, IF curl string passed in is not null:

3.2.2.1 reuse ParseCurl() (from clients/common/curlUtil.c) to get the pieces of the Curl file.

3.2.2.2 Need to further parse bcaps field to grab language field

3.2.2.3 Use bcaps field to map the hash value to the set of symbols out of the metafile (if it's "00", do a no-op)

3.2.2.3.1 On a template update/create, we have the physical symbols to be used in the page generator protocol b/c the metafile entry is part of what cmd is being told to update.

3.2.2.3.2 If not a template update (meaning, someone is using [CLEAR_CACHE] from within a template to flush the cache), we have to go to the metafile in the metafiles cache to lookup the symbols.

3.2.3 ELSE it's just a file, (the path was not a CURL, just a plain old file, e.g., index.html)

3.2.4 Call the function (generatePage()) to generate a request to the page generator and save a successful request to a tempfile

3.2.5 Perform rename from temp file to old filename

3.3 GeneratePage() needs to be added

3.3.1 Functionally equivalent to clients/common/generatePageImpl.cpp

3.3.2 Input: (*this is only for the ctld page generator*)

3.3.2.1 Need the filename (CURL, including template path, or plain file) modulo the docroot path, i.e., the URI

3.3.2.2 Needs the original Curl segment, if it is a CURL (pointer that went into TakeAction())

3.3.2.3 If it exists, needs the Content id out of the CURL

3.3.3 Output:

3.3.3.1 Two filenames: from and to

3.3.3.2 If there was an error along the way return the filename and its backup equivalent

3.3.3.3 Else return the temp filename generated and the original filename

3.3.4 Create an HTTP request to the page generator (*using clients/mt_nsapi/emulate404.c as a model, refer to http://hydra.vignette.com:8080/Interfaces/DOCUMENTS/Architecture_Info/ctld-protocol.html for explanation of the settings*)

3.3.4.1 Specify a POST to '/'

3.3.4.2 Add the HTTP header stuff

3.3.4.2.1 Our content-type is application/x-www-form-urlencoded

3.3.4.2.2 Don't forget the content-length that is the size of the entity body (following the double newline that separates headers from post bodies)

3.3.4.3 Append the following in URI encoding format (name=value&name2=value2...)

3.3.4.3.1 socket=true

3.3.4.3.2 cache=auto

3.3.4.3.3 if there was a curl, curl=/0,...

3.3.4.3.4 for all bcaps symbols, the format is browser=<symbol>&browser=<symbol>&...

3.3.4.3.5 path=<templatepath>

3.3.4.3.6 content-type=text/html (unless we have CMD_MIMETYPE setting, in which case we have to request for a mapping) really what's needed is test/html or application/octet-stream the page gen only cares about it for the purposes of inserting the magic string..., so here we can fudge. By this we mean if the generated page ends in htm or html (notice the lack of . so shtml can be caught as well), then specify text/html, else use application/octet-stream in case it's a graphic or some other file that doesn't get the magic string prepended.

3.3.4.3.7 For personalization settings, we need to pass the appropriate name-val pairs used by the CURL and COMPONENT commands. (autourl-tracking, etc.)

3.3.4.3.8 Add pzn_url_exclude from config info

3.3.4.3.9 template=<template id> (*this is optional*)

3.3.4.3.10 oid=<content id>

3.3.5 Number of simultaneous page generator requests should be throttled (to avoid overburdening)

3.3.5.1 Counting semaphore on page generator requests

3.3.6 Call the page generator

3.3.7 If the response is a 200 only (i.e., if it is an OK response), save it to a temp file, and return the temp filename

3.3.8 Else, return the orig filename and backup version of it

3.3.9 The threading model in the cache manager is such that per path a thread is performing the flush, so nothing too fancy needs to happen on the temp filename in the same directory as where you're performing the rename (that also limits the clashing possibilities)

4 Other thoughts

4.1 Use ACE when appropriate

4.2 Config settings added to this:

4.2.1 Number of simultaneous regeneration requests to perform

4.2.2 Time window for what determines old versus new; think of the setting is inclusive for new, and make it seconds so comparisons are easy against st_atime

4.3 What's needed in the short term (i.e. for the customer in question):

4.3.1 Don't worry about browser caps (template variations); the customer in question isn't using them

4.3.2 Don't worry about how a time window can be specified in the flush protocol (in fact, no changes to the incoming flush request protocol should be done)

4.3.3 Keep the HTTP headers simple for now.

4.3.4 Do what makes sense as in all cases, but think in terms of this code lasting for awhile

5 Issues

5.1 In 5.5, Cobra issue: *Need to switch on template type to know which page generator (and which protocol) to use to make the request.*

5.2 Need to optimize, file checking to leverage metafile data.

5.3 RENAME and DELETE aren't granular enough for our actions:

5.3.1 We need to distinguish between template destruction ("DESTROY") requiring cache cleanup versus don't bother with a backup file for "old" cached files ("DELETE")

5.4 Need to manage virtual hosting requests: at some point we need to know the domain prefix mapping, so we need the prefix piece...

5.5 Need to get server-side values for HTTP headers sent to page generator representing the Server-side CGI variables

5.6 CTLD configs must allow CMD to talk to it, so if you have allowable IPs and NAMEs restricted, this needs to be opened up to the CMD (or set of CMDs)

5.7 In the 5.x and above releases, new references will need to be added for CMD to continue to do this...

5.8 Do we care if it's for a component? Not yet...

X. RELATED PROCEEDINGS APPENDIX

Appellant believes that there are no related appeals or interferences.